CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- At time of the Action: Claims 1-15 and 18-22.
- After this Response: Claims 1-15 and 18-22.

Canceled or Withdrawn claims: none.

Amended claims: 1, 4, 5, 9, 11, 13, 18, and 21.

New claims: none.

CLAIMS:

1. **(CURRENTLY AMENDED)** A method for generating a delta between a first program binary and a second program binary, the method comprising the steps of:

obtaining a first control flow graph (CFG) representation of the first binary and obtaining a second CFG representation of the second binary;

comparing the first and second CFG representations to identify blocks (nominally matched blocks) that match in the first and second CFG representations, thereby identifying blocks (nominally unmatched blocks) in the second CFG representation that do not match in the first CFG representation, the comparing being based upon content of blocks being compared and local neighborhoods of blocks surrounding blocks being compared, wherein a local neighborhood of a particular block consists of blocks neighboring that block in a CFG representation, but less than all the blocks in that CFG representation;

determining edit-operations that merges the unmatched blocks into the first CFG representation so that first CFG representation is substantially identical to the second CFG representation;

producing a delta comprising the unmatched blocks and the edit-operations.

2. **(ORIGINAL)** A method for transmitting a delta comprising:

a method for generating a delta as recited in claim 1;

transmitting the delta over a network.

3. **(ORIGINAL)** A method for patching a copy of the first program binary, the method comprising:

a method for generating a delta as recited in claim 1;

patching the copy of the first program binary so that the copy is substantially identical to the second program binary, wherein the delta guides such patching.

4. **(CURRENTLY AMENDED)** A method as recited in claim 1, ~~wherein the comparing step further comprises matching blocks across the first and second CFG representations based, at least partially, upon content of blocks being compared and neighborhoods of blocks local to blocks being compared~~ wherein a local neighborhood of a particular block consists of those blocks immediately adjacent that block.

5. **(CURRENTLY AMENDED)** A method as recited in claim 1, ~~wherein the comparing step further comprises matching blocks across the first and second CFG representations based, at least partially, upon content of the blocks being compared and a local neighborhoods of blocks surrounding the blocks being compared,~~ wherein a local neighborhood of a particular block is augmented with a random sampling of blocks from a substantially large neighborhood of blocks surrounding ~~the block~~ that block.

6. **(ORIGINAL)** A computer-readable medium having embodied thereon a data structure, comprising a delta generated in accordance with the steps recited in claim 1.

7. **(ORIGINAL)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 1.

8. **(ORIGINAL)** A method for matching blocks between a first control flow graph (CFG) representation of a portion of a first program and a second CFG representation of a portion of a second program, the method comprising:

matching blocks between the first and second CFG representations based upon the content of the blocks;

detecting outliers, wherein outliers are blocks in the first CFG representation that do not match any block in the second CFG representation during the matching step;

computing a neighborhood of each block in the first and second CFG representations by performing a breadth first traversal;

removing the outliers from each neighborhood.

9. **(CURRENTLY AMENDED)** A method as recited in claim 8 further comprising:

computing labels for each block in first and second CFG representations based upon content of a block;

for each neighborhood computed in the computing step, forming a "d-label" for each block in a neighborhood based upon ~~the labels~~ labels of the blocks within the neighborhood;

attempting to match blocks between first and second CFG representations by comparing the d-labels of the blocks.

~~d-label matching blocks between first and second CFG representations based upon the d-label of the blocks.~~

10. **(ORIGINAL)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 8.

11. **(CURRENTLY AMENDED)** A method for matching procedures between a first control flow graph (CFG) representation of a portion of a first program and a second CFG representation of a portion of a second program, wherein a procedure comprises multiple blocks in a CFG representation, the method comprising:

computing a procedure-match-criterion for each procedure in the second CFG representation, where the procedure-match-criterion for a procedure in the second CFG representation represents the number of matching blocks between that procedure and a specified procedure in the first CFG representation;

matching procedures in the second CFG representation with the specified procedure in the first CFG representation based upon the procedure-match-criteria for the procedures in the second CFG representation.

12. **(ORIGINAL)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 11.

13. **(ORIGINAL)** A method for matching of blocks in a procedure of a first control flow graph (CFG) representation of a portion of a first program between an ostensibly matching procedure of a second CFG representation of a portion of a second program, the method comprising:

matching blocks between the first and second CFG representations based upon the content of the blocks;

computing successively smaller neighborhoods of each block in the first and second CFG representations via breadth first traversals;

for each neighborhood computed in the computing step, forming a "d-label" for each block in a neighborhood based upon ~~the labels~~ labels of the blocks within the neighborhood;

attempting to match blocks between first and second CFG representations by comparing the d-labels of the blocks.

~~d-label matching blocks between first and second CFG representations based upon the d-label of the blocks.~~

14. **(ORIGINAL)** A method as recited in claim 13, wherein at least one neighborhood computed in the computing steps is augmented with a random sampling of blocks in the complete representation of the neighborhood.

15. **(ORIGINAL)** A computer-readable medium having computer-executable instructions that, when executed by a computer, performs the method as recited in claim 13.

16. **(NOT ENTERED)**

17. **(NOT ENTERED)**

18. **(CURRENTLY AMENDED)** A patch data structure generated in accordance with the following acts:

providing a server computer in a communications with a communications network;

receiving input from a client computer by way of the communications network, the input providing a parameter indicative of a request for upgrading a copy of a first program binary to a match a second program binary;

retrieving a delta between a first program binary and the second program binary, wherein computing such delta comprises the steps of:

a) obtaining a first control flow graph (CFG) representation of the first binary and obtaining a second CFG representation of the second binary;

b) comparing the first and second CFG representations to identify blocks (nominally matched blocks) that match in the first and second CFG representations, thereby identifying blocks (nominally unmatched blocks) in the second CFG representation that do not match in the first CFG representation, the comparing being based upon content of blocks being compared and local neighborhoods of blocks surrounding blocks being compared, wherein a local neighborhood of a particular block consists of blocks neighboring that block in a CFG representation, but less than all the blocks in that CFG representation;

c) determining edit-operations that merges the unmatched blocks into the first CFG representation so that first CFG representation is substantially identical to the second CFG representation;

d) producing a delta comprising the unmatched blocks and the edit-operations;

generating the patch data structure as a function of the delta.

19. (ORIGINAL) A method for transmitting a patch data structure comprising transmitting a patch data structure as recited in claim 18 over a communications network.

20. (ORIGINAL) A method for patching a copy of the first program binary at a client computer, the method comprising patching the copy of the first program binary so that the copy is substantially identical to the second program binary, wherein a delta in a patch data structure as recited in claim 18 guides such patching.

21. **(CURRENTLY AMENDED)** A delta-generator system, comprising:

a comparator that is configured to compare a first control flow graph (CFG) representation of a first program binary and a second CFG representation of the second program binary for identifying blocks (nominally matched blocks) that match in the first and second CFG representations, thereby identifying blocks (nominally unmatched blocks) in the second CFG representation that do not match in the first CFG representation, the comparison being based upon content of blocks being compared and local neighborhoods of blocks surrounding blocks being compared, wherein a local neighborhood of a particular block consists of blocks neighboring that block in a CFG representation, but less than all the blocks in that CFG representation;

an edit-op determiner configured to determine the edit-operations that merges the unmatched blocks into the first CFG representation so that first CFG representation is substantially identical to the second CFG representation;

an output sub-system that is configured to produce a delta comprising the unmatched blocks and the edit-operations.

23. **(ORIGINAL)** A computer-readable medium having embodied thereon a data structure comprising a delta produced by the system as recited in claim 21.